REMARKS

Claims 1-7, 9-32, 60-76 and 78-86 were pending. With this Amendment, new claims 87-99 have been added while claims 18-19, 21, 23, 25-28, 60-76, 78-83, and 85-86 have been cancelled without prejudice. Applicants reserve the right to file these cancelled claims in one or more continuation applications. Upon entry of the present Amendment, claims 1-7, 9-17, 20, 22, 24, 29-32, 84 and 87-99 will be pending.

With this amendment claim 1 was amended as set forth in the following table, where exemplary support in the specification for these amendments is further provided.

Claim	Support in Specification
1. (Currently amended) A method, comprising: (A) obtaining, at a server, a plurality of e-mails from one or more remote client computers, wherein the plurality of e-mails are intended for distribution by the server to a plurality of respective remote destinations;	Paragraph 31; Figure 1
(B) creating, using one or more suitably programmed computers, a respective data node for each respective e-mail in said plurality of e-mails, wherein each respective data node includes (i) a pointer, wherein the pointer identifies ~~to~~ the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks a number of attempts made to deliver the respective e-mail;	Paragraph 32 'The data node includes information about the message being send; and may include the recipient, sender, data about the email, domain, a unique session identifier, visit count, and other information for Quality of Service ("QoS") guarantees and routing specifics. Note that the nodes are not emails themselves – rather they are just pointers to the emails stored in memory.'; and Paragraph 41 'Each time the message delivery fails, the "visit count" is incremented.'

Claim	Support in Specification
(C) processing, using one or more suitably programmed computers, the plurality of data nodes ~~solely~~ within a ~~non persistent~~ non-persistent storage comprising a plurality of queues, wherein each queue in the plurality of queues corresponds to a specific domain, without requiring that information indicative of the e-mails be written to and then read from persistent storage during the processing (C) of the data nodes, wherein said processing (C) comprises, for each respective data node:	Paragraph 31
(i) determining a destination domain of the respective data node;	
(ii) adding the respective data node to a queue in the non-persistent storage corresponding to the destination domain of the respective data node when the queue exists in the plurality of queues; ~~and~~	Paragraph 29
(iii) creating a queue in the non-persistent storage corresponding to the destination domain and adding the respective data node to the created queue when the queue does not exist in the plurality of queues, wherein	Paragraph 35
said processing (C) further comprises a method comprising: (a) selecting a ~~first~~ respective queue in the plurality of queues that contains data nodes;	

Claim	Support in Specification
(b) retrieving e-mails corresponding to each of the data nodes in the ~~first~~ respective queue;	
(c) finding a remote server corresponding to the destination domain of the respective queue;	Paragraph 40, where the remote server in the example illustrated in paragraph 40 is an SMTP server
(d) sending each of the retrieved e-mails corresponding to each of the data nodes in the ~~first~~ respective queue to the remote server corresponding to the [[a]] destination domain of the ~~first~~ respective queue, wherein the sending (d) comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the respective queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail,	Paragraph 41 and 45
wherein, when a delivery failure message is received for a first e-mail in the retrieved e-mails at a time after said sending (d), said processing (C) further comprises: (i) pushing the data node corresponding to the first e-mail back onto the respective queue, and (ii) incrementing the visit count in the data node corresponding	Paragraph 41

Claim	Support in Specification
to the first e-mail to account for the failed delivery of the first e-mail; and	
(e) extinguishing the ~~first~~ respective queue when all e-mails in the respective queue have been delivered without receiving a delivery failure message.	Paragraph 41

With this Amendment, claim 17 was amended for clarity to recite storing recovery information about a state of processing of the plurality of e-mails to persistent storage, wherein said recovery information comprises less than the entirety of each of the e-mails in the plurality of e-mails. Support for this amendment to claim 17 is found in paragraph 54 of the specification.

With this Amendment, claim 84 was amended in the same manner as claim 1 set forth above. Claim 84 was further amended to recite "creating a queue corresponding to the destination domain and adding the respective data node to the created queue when the queue does not exist" for clarity and to correct for antecedent basis.

New claim 88 is a computer system claim that is supported by the specification on the same basis that claim 84 is supported.

New claims 89-90 are supported by paragraph 53 of the specification.

New claims 91-93 are supported by paragraph 37 of the specification.

New claims 94-96 are supported by paragraph 35 of the specification.

New claims 97-99 are supported by paragraph 31 of the specification.

No new matter has been introduced by way of these amendments to the claims.

In the April 24, 2009 Office Action, the Examiner:

- rejected claims 1, 60 and 73 under 35 U.S.C. § 101 for allegedly failing to positively recite the statutory class to which they are tied or transform underlying subject matter to a different state or thing;

- rejected claims 1 and 84 under 35 U.S.C. § 112, second paragraph, as being indefinite for reciting "…creating a queue corresponding to the destination domain and adding the respective data node to the created queue when the queue does not exist";
- rejected claims 1-7, 18-21 and 23-25 under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. publication number US2003/0028580 to Kucherawy (hereinafter, "Kucherawy") in view of United States Patent No. 6,249,807 to Shaw (hereinafter, "Shaw");
- rejected claims 9-15, 17, 22 and 26-31 under 35 U.S.C. § 103(a) as allegedly being unpatentable over the combination of Kucherawy and Shaw in further view of European Patent Application EP 0 491 367 A2 to Richard E. Batchelor (hereinafter, "Batchelor");
- rejected claims 60-67, 73-74, 78-81 and 85-86 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Kucherawy;
- rejected claims 68-71, 75-76 and 82 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Kucherawy in view of Batchelor; and
- rejected claims 16, 32, 72 and 83 under 35 U.S.C. § 103(a) as allegedly being unpatentable over the combination of Kucherawy, Shaw, Batchelor and further in view of United States Patent Publication No. US2002/0143885 to Ross (hereinafter, "Ross").

APPLICANTS' INTERVIEW SUMMARY

Applicants thank Examiner Ryan J. Jakovak for the courtesies extended during the in person interview on December 15, 2009, hereinafter "the Interview," with Applicants' representative Brett Lovejoy inventor Tim McQuillen, and representative for Assignee StrongMail, Shekhar Yadov. During the Interview, the rejection of the claims over the cited art was discussed. Possible claim amendments were also discussed. It was agreed that a Supplemental Response to the April 24, 2009 Office Action would be filed by December 22, 2009 and that Examiner Jakovak would consider the Supplemental Response rather than Applicants' response filed on October 26, 2009.

THE 35 U.S.C. § 101 REJECTION SHOULD BE WITHDRAWN

The Examiner has rejected claims 1, 60 and 73 under 35 U.S.C. § 101 for allegedly failing to positively recite the statutory class to which they are tied or transform underlying subject matter to a different state or thing. Applicants note that the rejection is moot as it applied to claims 60 and 73 in view of the cancellation of these claims without prejudice.

Applicants have amended claim1 to specify obtaining a plurality of e-mails at a server, creating data nodes using one or more suitably programmed computers, and processing using one or more suitably programmed computers. Accordingly, Applicants respectfully request that the 35 U.S.C. § 101 rejection be withdrawn.

THE 35 U.S.C. § 112, SECOND PARAGRAPH, REJECTION SHOULD BE WITHDRAWN

The Examiner has rejected claims 1 and 84 under 35 U.S.C. § 112, second paragraph, as being indefinite for reciting "…creating a queue corresponding to the destination domain and adding the respective data node to the created queue when the queue does not exist". Applicants traverse the rejection. Claim 1 and claim 84, as amended, specify that the data node is added to the created queue. Thus claim 1 and claim 84, as amended, do not specify adding a data node to a queue that does not exist. Accordingly, Applicants respectfully request that the 35 U.S.C. § 112, second paragraph, rejection be withdrawn.

THE REJECTION OF CLAIMS 1-7, 18-21 AND 23-25 UNDER 35 U.S.C. § 103(A) SHOULD BE WITHDRAWN

The Examiner has rejected claims 1-7, 18-21 and 23-25 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Kucherawy in view of Shaw. Applicants note that the rejection is moot with respect to claims 18, 19, 21, 23 and 25 in view of the cancellation of these claims without prejudice. Applicants traverse the rejection as it applies to claims 1-7, 20, and 24.

Claim 1 requires the creation of a data node for each e-mail in a plurality of e-mails, where each data node includes (i) a pointer, wherein the pointer identified the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks the number of attempts made to deliver the respective e-mail. Kucherawy does not disclose this requirement of claim 1. Kucherawy discloses both generic destination and specific destination E-mail queues. Kucherawy at [0044]. The entirety of each E-mail in each of these queues in Kucherway is kept in non persistent storage. See, for example, the data structure for a queue disclosed in paragraph [0093] of Kucherawy. Line 46 of the data structure disclosed in paragraph [0093] of Kucherawy specifies:

```
46: Vector q_msgs; / * queued messages */
```

This is possible in Kucherawy because the generic destination and specific destination E-mail queues of Kucherawy are the queues of a mass mail accelerator (MMA), not a high performance mail transfer agent (MTA). Thus, because the entirety of each E-mail in the Kucherawy queues is kept in non persistent storage, both the generic destination and specific destination E-mail queues of Kucherawy do not disclose, teach or suggest the creation of a data node for each e-mail in a plurality of e-mails, where each data node includes a pointer to the corresponding respective e-mail in persistent storage as required by claim 1.

On page 3 of the April 24, 2009 Office Action, the Examiner contends that Kucherawy contemplates using pointers to the retry queue as well as reference handles for duplicate messages. Applicants respectfully point out that paragraph [0089] of Kucherawy explains that the reason that the generic destination and specific destination E-mail queues contain a pointer to a retry queue is so that the generic destination and specific destination queues know where to discard an undeliverable e-mail:

> [0089] A queue can be configured to pass messages it cannot complete to some other queue for special handling. This might be caused, for example, by a total network failure between the MMA and a particular destination, such as AOL.com or hotmail.com, as in previous examples. Where this is the case, the queue thread which decides it is unable to complete the delivery will hand the message to its designated "retry" queue, and that queue will attempt to deliver the message using the MTAs and other features available to it. If the initial queue thread has no "retry" queue assigned, the message fails completely, an

> error is logged, and the message is discarded (or in safe mode, the SMTP submission into the MMA fails).
>
> Paragraph [0089] of Kucherawy.

Thus, given paragraph [0089], Kucherawy clearly contemplates passing the entirety of an undeliverable e-mail to the retry queue. Moreover, there is no disclosure in Kucherawy that the retry queue requires the creation of a data node for each e-mail in a plurality of e-mails, where each data node includes a pointer to the corresponding respective e-mail in persistent storage. Thus, the pointer to a retry queue in the Kucherawy generic destination and specific destination E-mail queues fails to disclose, teach or suggest the creation of a data node for each e-mail in a plurality of e-mails, where each data node includes a pointer to the corresponding respective e-mail in persistent storage.

The reference handles for duplicate messages in Kucherawy that were identified by the Examiner on page 3 of the April 24, 2009 Office Action also fails to disclose teach or suggest the creation of a data node for each e-mail in a plurality of e-mails, where each data node includes a pointer to the corresponding e-mail in <u>persistent</u> storage because it is clear that the entirety of each e-mail in the Kucherawy generic destination and specific destination E-mail queues is within non persistent storage. Therefore, the Kucherawy handles to duplicate messages are handles to <u>non persistent</u> storage not <u>persistent</u> storage as required by claim 1.

Kucherawy also fails to disclose, teach or suggest extinguishing a first queue as part of the act of processing data nodes in non persistent storage in which (i) a first queue that contains data nodes is selected, (ii) e-mails corresponding to each of the data nodes in the first queue are retrieved, (iii) each of the retrieved e-mails corresponding to each of the data nodes in the first queue is sent to a destination domain of the first queue, and (iv) the first queue is extinguished as required by claim 1. In fact, Kucherawy teaches exactly the opposite. Kucherawy teaches that the generic destination and specific destination E-mail queues are permanent. For example, Kucherawy discloses:

> Which queues are created is entirely dependent on the configuration which gives the customer-user (e.g., system administrator) the ability to tailor or tune

> for a given situation. If, for example, the system administrator knows that about 60% of outgoing e-mail for his or her company is going to AOL, then the system administrator would set up an AOL-specific queue, with corresponding resources.
>
> Paragraph [0084] of Kucherawy

On page 5 of the April 24, 2009 Office Action, the Examiner contends that paragraph [0105] of Kucherawy discloses the extinguishing of a queue. This is not the case. A Kucherawy queue may go to sleep while waiting for more messages to process. However, a Kucherawy queue is not extinguished as required by Applicants' claim 1. See for example, paragraph [0041] of Applicants' specification:

> [0041] Once all of the messages in the queue have been removed, then the queue is removed from the memory map, or extinguished, at 445.
>
> Applicants' specification as filed at [0041].

As an additional matter, the Kucherawy retry queue and generic destination queues are not comparable to Applicants' queues because each of Applicants' queues corresponds to a particular destination domain whereas the Kucherawy retry queue and generic destination queues do not.

As an additional matter, Kucherawy fails to disclose, teach or suggest sending each of a plurality of retrieved e-mails corresponding to each of the data nodes in a first queue to a remote server corresponding to the destination domain of the first queue, where the sending comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the first queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail as required by claim 1.

Furthermore, Kucherawy fails to disclose, teach or suggest pushing a data node corresponding to an e-mail back onto a first queue corresponding to the destination domain of the first queue and incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the first e-mail when a delivery failure message is received for a first e-mail in the retrieved e-mails at a time after the sending of the e-mail. In fact, Kucherawy teaches away from such an approach. Kucherawy uses a retry queue, rather than pushing e-mails back to the original queue.

Shaw fails to remedy the above-identified deficiencies in Kucherawy. As a preliminary matter, the queues of Shaw are not used to send e-mails to destination domains. Shaw refers to a system that receives e-mails from destination domains. Shaw does not send e-mails from queues to destination domains. As such, the only relevance that Shaw has to Applicants' claims or to Kucherawy is that Shaw *might* receive e-mails sent from an MTA in accordance with Applicants' claims or in accordance with the teachings of Kucherawy. As such, the combination of Kucherawy and Shaw is not a fair combination and such combination would not have been contemplated by one of skill in the art.

More specifically, Shaw does not disclose, teach or suggest the creation of a data node for each e-mail in a plurality of e-mails, where each data node includes (i) a pointer, wherein the pointer identified the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks the number of attempts made to deliver the respective e-mail. Claim 1 requires that this plurality of e-mails be intended for distribution to a plurality of remote destinations. On page 3 of the April 24, 2009 Office Action, the Examiner contends that Shaw, column 9, lines 20-30, discloses a queue in which each data node includes a pointer to the corresponding e-mail in persistent storage. The disclosure at column 9, lines 20-30, of Shaw has no relevance to Applicants claims because all that it discloses is a mailbox of received e-mail messages. As such, the plurality of e-mails disclosed in this passage in Shaw is not intended for distribution to a plurality of respective remote destinations. In fact, the word "queue" is used in this passage to simply suggest that the mailbox will list the pointers to the received e-mails in, for example, the order the e-mails had been received.

Shaw, like Kucherawy, also fails to disclose, teach or suggest sending each of a plurality of retrieved e-mails corresponding to each of the data nodes in a first queue to a remote server corresponding to the destination domain of the first queue, where the sending (vii) comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the first queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail as required by claim 1.

Furthermore Shaw, like Kucherawy, fails to disclose, teach or suggest pushing a data node corresponding to an e-mail back onto a first queue corresponding to the destination domain of the first queue and incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the first e-mail when a delivery failure message is received for a first e-mail in the retrieved e-mails at a time after the sending of the e-mail.

Claims 2-7, 20, 24 and 25 depend from claim 1 and thus are patentable over the combination of Kucherawy and Shaw for at least the same reasons. Accordingly Applicants respectfully request that the 35 U.S.C. § 103 rejection of claims 1-7, 18-21 and 23-25 be withdrawn.

THE REJECTION OF CLAIMS 9-15, 17, 22 AND 26-31 UNDER 35 U.S.C. § 103(A) SHOULD BE WITHDRAWN

The Examiner has rejected claims 9-15, 17, 22 and 26-31 under 35 U.S.C. § 103(a) as allegedly being unpatentable over the combination of Kucherawy and Shaw in further view of Batchelor. Applicants note that the rejection is moot with respect to claims 26-28 in view of the cancellation of these claims. Applicants traverse the rejection as it applies to claims 9-15, 17, 22 and 29-31.

Claims 9-15, 17, 22 and 26-31 each ultimately depend from claim 1. As discussed hereinabove, claim 1 requires creation of a data node for each e-mail in a plurality of e-mails, where each data node includes (i) a pointer, wherein the pointer identified the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks the number of attempts made to deliver the respective e-mail. Claim 1 also requires extinguishing a first queue as part of the act of processing data nodes in non persistent storage in which (i) a first queue that contains data nodes is selected; (ii) e-mails corresponding to each of the data nodes in the first queue are retrieved; (iii) each of the retrieved e-mails corresponding to each of the data nodes in the first queue is sent to a destination domain of the first queue; and (iv) the first queue is extinguished as required by claim 1. Claim 1 further requires sending each of a plurality of retrieved e-mails corresponding to each of the data nodes in a first queue to a remote server corresponding to the destination domain of the first queue, where the sending comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the first queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail as required by claim 1. Claim 1 further requires pushing a data node corresponding to an e-mail back onto a first queue corresponding to the destination domain of the first queue and incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the first e-mail when a delivery failure message is received for a first e-mail in the retrieved e-mails at a time after the sending of the e-mail.

As discussed hereinabove, the combination of Kucherawy and Shaw fails to disclose, teach or suggest these claim limitations.

Batchelor does not disclose, teach or suggest creation of a data node for each e-mail in said plurality of e-mails, where each data node includes a pointer to the corresponding e-mail in persistent storage. Batchelor fails to remedy the deficiencies in Kucherawy and Shaw. Batchelor discloses only a single queue, not a plurality of queues. This single queue is queue 16, which is disclosed in Figures 1 and 2 of Batchelor. The Batchelor queue contains requests. See Figures 1 and 2 of Batchelor. The format of such requests is set forth on page 7, lines 15-28, of Batchelor. There it is specified that each Batchelor request includes a data field that contains information about the message associated with the request (char data_field[150]; / data string). See, for example, page 7, lines 42-50, of Batchelor. As noted in the last sentence of this passage on page 7 of Batchelor, the first part of the data string stored in data_field[150] contains the name of an e-mail program that is invoked to carry out the processing of the request (bound unit) and the second part of the data string stored in data_field[150] is the character string that contains information which is passed to the named bound unit. As explained on page 9, lines 15-37, of Batchelor, the fourth parameter (argv[4]) within this information that is passed to the named bound unit of the request is the data string message provided by the requestor. Clearly, this data string message is in persistent memory and is not a pointer to a corresponding e-mail in persistent storage as required in claim 1.

Batchelor does not disclose, teach or suggest extinguishing a first queue as part of the act of processing data nodes in non persistent storage in which (i) a first queue that contains data nodes is selected; (ii) e-mails corresponding to each of the data nodes in the first queue are retrieved; (iii) each of the retrieved e-mails corresponding to each of the data nodes in the first queue is sent to a destination domain of the first queue; and (iv) the first queue is extinguished as required by claim 1. Moreover, based on the prior discussion of the Batchelor queue, it is clear that Batchelor does not disclose, teach or suggest extinguishing its only queue (queue 16). Thus, Batchelor does not disclose teach or suggest processing data nodes in non persistent storage in which (i) a first queue that contains data nodes is selected; (ii) e-mails corresponding to each of the data nodes in the first queue are retrieved; (iii) each of the retrieved e-mails corresponding to each of the data nodes in the first queue is sent to a destination domain of the first queue; and (iv) <u>the first queue is extinguished</u> as required by claim 1.

Batchelor does not disclose, teach or suggest additional claim limitations required by claim 1, as amended. Batchelor, like Kucherawy and Shaw, also does not disclose teach or suggest sending each of a plurality of retrieved e-mails corresponding to each of the data nodes in a first queue to a remote server corresponding to the destination domain of the first queue, where the sending comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the first queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail as required by claim 1. Batchelor, like Kucherawy and Shaw, also does not disclose teach or suggest pushing a data node corresponding to an e-mail back onto a first queue corresponding to the destination domain of the first queue and incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the first e-mail when a delivery failure message is received for a first e-mail in the retrieved e-mails at a time after the sending of the e-mail as required by claim 1.

In addition to the foregoing reasons, certain of the claims rejected over the combination of Kucherawy, Shaw, and Batchelor in point 7 of the April 24, 2009 Office Action are patentable over this combination for the additional, independent reasons set forth hereinbelow.

Claim 9. Claim 9 requires selection of a first queue which has the greatest number of the e-mails within the queue. In point 7 of the April 24, 2009 Office Action, the Examiner contends that Kucherawy and Shaw do not expressly disclose this feature. However, the Examiner contends that Batchelor does teach this feature. Applicants disagree. As a preliminary matter, Batchelor discloses only a single queue, not a plurality of queues. This single queue is queue 16, which is disclosed in Figures 1 and 2 of Batchelor. Thus, it is not possible for Batchelor to disclose the feature of selecting a first queue which has the greatest number of the e-mails within the queue.

Moreover, the Examiner's reliance in the April 24, 2009 Office Action on column 2, lines 50-57, of Batchelor to disclose this feature is misplaced. As explained on page 2, lines 46-51, of Batchelor, a queue manager separately stores a table containing an entry for each possible e-mail destination. This feature is disclosed in greater detail in Section C.3.2 of Batchelor, beginning on page 10, line 55. Each destination (node 24 in Figure 1; destinations 48 / DCFs 42 in Figure 2) has a set of "windows" in time during each 24 hour day during which requests having the corresponding priority may be executed to the given destination. As explained on page 11, lines 7-11, of Batchelor , each "window" has associated with it an "economic quantity" which specifies, for that destination, window and priority, the minimum number of requests that should be executed for optimum use of the destination and communications link resources. Queue handler 18 will hold all results for a given destination and of a given priority until the number of requests awaiting execution is equal to or greater than the "economic quantity"; when that number of pending requests reaches the "economic quantity", all requests will be executed. Thus, the Batchelor "windows" and their associated economic quantity information are used as a gate to tell queue handler 18 when messages may be sent to the destinations associated with the "windows". As such, the "windows" serve as binary gates, either accepting messages or not. They are not used to select a queue that has the greatest number of E-mails as required by claim 9. So, for example, there can be multiple windows for multiple destinations that are open at the same time and the Batchelor queue manager 18 can send messages to each of these destinations at the same time without any need to select a destination among the possible destinations for which the queue 16 holds the most mail.

In fact, the Batchelor "window" **teaches away** from the feature of selecting a first queue which has the greatest number of e-mails within the queue. For instance, it is possible, at any given instance of time, for there to be more e-mails in the single Batchelor queue 16 bound for a particular destination 24 (node; Batchelor Figure 1). Yet, the Batchelor queue manager 18 is barred from sending messages to this destination when the delivery criteria file (DCF) 42 for this destination indicates that the there is no open "window" at the given instance of time. Furthermore, even if the DCF 42 for this destination indicates that there is an open window at the given instance of time, the Batchelor queue manager 18 is barred from sending messages to this destination when the messages in the queue intended for the destination, although being great in number, do not have the priority value that matches the priority specified by the window for the destination at the given instance in time.

Claim 10. Claims 10 specifies selecting a first queue that has existed for the greatest period of time. The Examiner contends that Batchelor, column 4, lines 9-15, discloses this feature. Applicants disagree. As a preliminary matter, as discussed hereinabove with respect to the rejection of claim 9, Batchelor discloses only a single queue 16 (see Figures 1 and 2 of Batchelor). As further disclosed in Figures 1 and 2 of Batchelor, this single queue comprises a plurality of requests 22. These requests are not queues. What column 4, lines 9-15, of Batchelor discloses is nothing more than the basic feature that these requests (*e.g.*, e-mails) are executed in the order in which they were received from users. As such, this basic feature does not disclose, teach or suggest what is required of Applicants' claim 10.

Claim 17. Claim 17, as amended, requires the storage of recovery information about a state of procession of a plurality of e-mails to persistent storage, where the recovery information comprises less than the entirety of **each of the e-mails** in the plurality of e-mails. On page 8 of the April 24, 2009 Office Action, the Examiner contends that the combination of Kucherawy and Shaw discloses this feature. For clarity, Applicants have amended claim 17 to specify that the recovery information stores information about each of the e-mails, where the recovery information comprises less than the entirety of each of the e-mails. The combination of Kucherawy and Shaw does not disclose, teach or suggest this feature. For example, paragraphs [0089 - 0090] of Kucherawy merely discloses passing on the entirety of an undeliverable e-mail to a "retry" queue, not a portion of the e-mail (*e.g.*, a non persistent pointer to the e-mail stored in persistent storage) as claim 17, as amended, requires.

Accordingly, for the above-identified reasons, Applicants respectfully request that the 35 U.S.C. § 103 rejection of claims 9-15, 17, 22 and 26-31 be withdrawn.

THE REJECTION OF CLAIMS 60-67, 73-74, 78-81 AND 85-86 UNDER 35 U.S.C. § 103(A) SHOULD BE WITHDRAWN

The Examiner has rejected claims 60-67, 73-74, 78-81 and 85-86 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Kucherawy. Applicants note that the rejection is moot with respect to these claims in view of their cancellation without prejudice. Accordingly Applicants respectfully request that the 35 U.S.C. § 103 rejection of claims 60-67, 73-74, 78-81 and 85-86 be withdrawn.

THE REJECTION OF CLAIMS 68-71, 75-76 AND 82 UNDER 35 U.S.C. § 103(A) SHOULD BE WITHDRAWN

The Examiner has rejected claims 68-71, 75-76 and 82 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Kucherawy in view of Batchelor. Applicants note that the rejection is moot with respect to these claims in view of their cancellation without prejudice. Accordingly Applicants respectfully request that the 35 U.S.C. § 103 rejection of claims 68-71, 75-76 and 82 be withdrawn.

THE REJECTION OF CLAIMS 16, 32, 72 AND 83 UNDER 35 U.S.C. § 103(A) SHOULD BE WITHDRAWN

The Examiner has rejected claims 16, 32, 72 and 83 under 35 U.S.C. § 103(a) as allegedly being unpatentable over the combination of Kucherawy, Shaw, Batchelor and Ross. Applicants note that the rejection is moot with respect to claims 72 and 83 in view of their cancellation without prejudice. Applicants traverse the rejection as it applies to claims 16 and 32.

Claims 16 and 32 depend from claim 1. As discussed hereinabove with respect to the rejection of claim 1 over Kucherawy and Shaw, and claim 9 over the combination of Kucherawy, Shaw, and Batchelor, none of these references disclose, teach or suggest the following requirements of claim 1:

(A) creation of a data node for each e-mail in said plurality of e-mails, where each data node includes a pointer to the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks the number of attempts made to deliver the respective e-mail,

(B) extinguishing a first queue as part of the act of processing data nodes in non persistent storage in which (i) a first queue that contains data nodes is selected; (ii) e-mails corresponding to each of the data nodes in the first queue are retrieved; (iii) each of the retrieved e-mails corresponding to each of the data nodes in the first queue is sent to a destination domain of the first queue; and (iv) the first queue is extinguished,

(C) sending each of a plurality of retrieved e-mails corresponding to each of the data nodes in a first queue to a remote server corresponding to the destination domain of the first queue, where the sending comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the first queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail, or

(D) pushing a data node corresponding to an e-mail back onto a first queue corresponding to the destination domain of the first queue and incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the first e-mail when a delivery failure message is received for a first e-mail in the retrieved e-mails at a time after the sending of the e-mail as required by claim 1.

Ross, which discloses only basic information about storing e-mails in a queue 326, also fails to disclose, teach or suggest these features. Thus, claims 16 and 32 are fully patentable over any combination of Kucherawy, Shaw, Batchelor and Ross.

Accordingly Applicants respectfully request that the 35 U.S.C. § 103 rejection of claims 16, 32, 72 and 83 be withdrawn.

CONCLUSION

Applicants respectfully request that the above-mentioned amendments and remarks be entered and made of record in the file history of the subject application. It is believed that no fees are due in connection with the filing of this amendment. However, should the Patent

Office determine otherwise, please charge the required fee to Jones Day deposit account no. 50-3013, referencing CAM No. 687465-999003.

Respectfully submitted,

Date: December 22, 2009

/ Brett Lovejoy / 42,813

Brett Lovejoy (Reg. No.)

JONES DAY
222 East 41st Street
New York, New York 10017-6702
Phone: (415) 875-5744